Expert Systems with Applications 39 (2012) 2035–2046

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A multi-layer text classification framework based on two-level representation model

Jiali Yun, Liping Jing *, Jian Yu, Houkuan Huang

School of Computer and Information Technology, Beijing Jiaotong University, China

ARTICLE INFO

Keywords:
Text classification
Text representation
Multi-layer classification
Wikipedia
Semantics

ABSTRACT

Text categorization is one of the most common themes in data mining and machine learning fields. Unlike structured data, unstructured text data is more difficult to be analyzed because it contains complicated both syntactic and semantic information. In this paper, we propose a two-level representation model (2RM) to represent text data, one is for representing syntactic information and the other is for semantic information. Each document, in syntactic level, is represented as a term vector where the value of each component is the term frequency and inverse document frequency. The Wikipedia concepts related to terms in syntactic level are used to represent document in semantic level. Meanwhile, we designed a multi-layer classification framework (MLCLA) to make use of the semantic and syntactic information represented in 2RM model. The MLCLA framework contains three classifiers. Among them, two classifiers are applied on syntactic level and semantic level in parallel. The outputs of these two classifiers will be combined and input to the third classifier, so that the final results can be obtained. Experimental results on benchmark data sets (20Newsgroups, Reuters-21578 and Classic3) have shown that the proposed 2RM model plus MLCLA framework improves the text classification performance by comparing with the existing flat text representation models (Term-based VSM, Term Semantic Kernel Model, Concept-based VSM, Concept Semantic Kernel Model and Term + Concept VSM) plus existing classification methods.

1. Introduction

Text categorization is one of the most common themes in data mining and machine learning fields. The task of text categorization is to build a classifier based on some labeled documents and to classify the unlabeled documents into the prespecified categories. As we know, text data is unstructured, thus it cannot be directly processed by the existing classification algorithms. In order to structure text data, Bag of Words (BOW) model (Yates & Neto, 1999) (i.e. term-based Vector Space Model (VSM)) is popularly used here. In term-based VSM, a document is represented as a feature vector which consists of the words in all of the documents. Term-based VSM has been widely applied to text categorization due to its simplicity and good performance. However, it has two main drawbacks: (1) it does not consider the semantic relatedness between words, i.e., two words with similar meanings are treated as irrelevant features in term-based VSM. (2) The same word in different contexts cannot be differentiated if they have different meanings (Feldman & Sanger, 2007). In other words, term-based VSM does not cover the semantic information, which limits its performance in many text mining tasks, e.g., classification, clustering and information retrieval.

Attempts have been made in the literatures to introduce semantic information to text representation with the aid of background knowledge bases, such as WordNet,[1] ODP[2] and Wikipedia.[3] Some approaches were proposed to integrate semantic information into representation model by identifying concepts related to words, such as concept-based VSM (Huang, Milne, Frank, & Witten, 2008). However, there are many problems in concept-based VSM. The main problem is how to correctly find the appropriate related concepts for each word in the corresponding context. Furthermore, the existing background knowledge bases do not cover concepts related to all terms. To overcome these drawbacks, a good solution is utilizing both term and concept information. In the literature, two major methods were adopted. One is to merge term vector and concept vector into a flat longer vector (term + concept vector) via add strategy or replace strategy (Hotho, Staab, & Stumme, 2003). The other is to combine term and concept information during the learning process by linearly integrating the document similarity based on term vector and similarity based on concept vector, in this case, the importance of term or concept can be controlled via a parameter

* Corresponding author.
E-mail address: lpjing@bjtu.edu.cn (L. Jing).

[1] http://wordnet.princeton.edu/.
[2] http://www.dmoz.org/.
[3] http://www.wikipedia.org.

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.08.027

(Hu, Zhang, Lu, Park, & Zhou, 2009). Even though such integrated representation models were experimentally shown to improve the performance of text categorization, how to sufficiently apply syntactic and semantic information is still an open problem. Moreover, these methods use flat feature representation model which can not consider the spatial distribution of terms and concepts (Chow & Rahman, 2009).

In order to make use of syntactic and semantic information, we propose a two-level representation model (2RM). Different from the existing flat document representation models, 2RM represents document in a two-level vector space containing syntactic (term) and semantic (related concept) information respectively. The syntactic level represents each document as a term vector, and the component records tf-idf value of each term. The semantic level represents document with Wikipedia concepts related to terms in syntactic level. A context-based method is adopted to identify the appropriate Wikipedia concepts and then build the semantic level vector space. In other words, whether a concept is related to one term in a document depends on the extent to which this concept is similar to the concepts related to other terms in this document. Also, such relatedness is used to weight the contribution of related concepts to categorization. Furthermore, a structure-based solution is proposed to fast identify the appropriate Wikipedia concept for long document, where the concept is only compared with the concepts related to other terms in a predefined window size (e.g., same paragraph) of the document. Experiments, in this paper, have shown that using nearest neighbor paragraphs as context can greatly reduce the computational complexity of constructing concept-based representation model than using full document as context at the cost of a little worse performance of classification.

Based on 2RM, we design a multi-layer classification (MLCLA) framework to analyze text data in a way of layer-by-layer. MLCLA framework includes three classifiers. Among them, two classifiers are applied on document represented in two levels respectively and independently. Based on the output of these two classifiers, each document is represented as two compressed vectors. The combined vector from the above two compressed vectors will be input to the third classifier to obtain the final results. MLCLA framework effectively keeps the primary information and reduces the influence of noise by compressing the original information, so that the proposed framework guarantees the quality of the input of the classifier. In addition, MLCLA can be implemented flexibly in series or in parallel according to the memory available and the required time. Therefore, it is more practical for dealing with large scale of documents collection.

The rest of the paper is organized as follows: Section 2 reviews related works on text representation with aid of background knowledge and classification techniques. Section 3 proposes the two-level representation model. Section 4 presents the MLCLA framework. Section 5 describes experiments and discusses results. Finally, we conclude the paper and give the future work in Section 6.

2. Related work

VSM is the most popular document representation model for text clustering, classification and information retrieval. In early literature, term-based VSM, representing one document as a term vector, was widely used. The weight of each term in a document is usually measured via two schemes: Binary (1 for term appearing in the document, 0 for not) and Term Frequency-Inverse Document Frequency (TF-IDF). However, both approaches only contain the literal information in document. Some methods were proposed to mine the underlying semantic structure in textual data, such as Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003) and Latent Semantic Indexing (LSI) (Deerwester, Dumais, Landauer, Furnas, & Harshman, 1990; Nouali & Blache, 2004). To some extent, these methods make up for the shortage of term-based VSM, but they cannot discover as much semantic information as described in text data only by analyzing syntactic information via statistic methods. Jing, Zhou, Ng, and Huang (2006) implicitly embedded the semantic information to document representation via kernel method by multiplying document-term tf-idf matrix and term similarity matrix, where the term similarity was computed based on WordNet.

Recently, as the available background knowledge bases grow, some researchers tried to use them to build concept-based VSM which can explicitly represented the semantic information of documents (Gabrilovich & Markovitch, 2007). The crucial step to build concept-based VSM is extracting concepts from knowledge base. Hotho et al. (2003) took the synonyms in Wordnet of each term as the related concepts. Although empirical results have shown this method was efficient in some cases, Wordnet is manually built and its coverage is far too restricted. Thus, many researches began to make use of Wikipedia, the largest electronic encyclopedia to date. Wang, Hu, Zeng, Chen, and Chen (2007) and Hu et al. (2008) constructed an informative thesaurus from Wikipedia so that the synonymy, polysemy, hyponymy, and associative relations between concepts can be explicitly derived. But they rely on an exact phrase matching strategy while this strategy is limited by the terms appearing in the documents and the coverage of Wikipedia concepts or article titles. Hu et al. (2009) built document-concept matrix through exact-match and relatedness-match which requires to compute the tf-idf value of term in the whole Wikipedia article collection. It is time consuming. Banerjee, Ramanathan, and Gupta (2007) treated the entire document as query strings to Wikipedia and associate the document with the top articles in the returned result list. Gabrilovich and Markovitch (2005, 2006, 2007) used machine learning techniques to map document to the most relevant concepts in ODP or Wikipedia by comparing the textual overlap between each document and article. However, its feature generation procedure requires high processing efforts, because each document needs to be scanned multiple times. Besides, it produced too many Wikipedia concepts for each document and filtering step further increases the processing time. Syed, Finin, and Joshi (2008) was interested in finding semantically related concepts which were also common to a set of documents. Huang et al. (2008), Huang, Milne, Frank, and Witten (2009) mapped candidate phrases in the given document to Wikipedia articles by leveraging an informative and compact vocabulary – the collection of anchor texts in Wikipedia. Our adopted method is more similar with Huang et al. (2009) used where Wikipedia's anchor text vocabulary is used to connect terms to Wikipedia articles. In this way the number of concepts in a document is no more than the number of terms. Meanwhile, different terms with the same meaning might be mapped to the same Wikipedia article because anchors linked to the same article are also often couched in different words.

Beside identifying the related concepts, weighting the concepts is also a vital technology to build concept-based VSM. Huang et al. (2009) used tf-idf strategy to weight concept and extended the concept relatedness mentioned in (Milne & Witten, 2008) to the similarity between documents. Wang et al. (2008) adopted a similar kernel method with Jing et al. (2006) to enrich document representation matrix, which replaced concept similarity matrix for the term similarity matrix. Concept similarity matrix was measured by taking account of synonyms, hyponyms and associative concepts in Wikipedia. However, These methods do not use the contextual semantic relatedness to change the concept weight. In this paper, concept weight is effected by the semantic relatedness between concept and the given document, which is equal to the

average relatedness between concept and other concepts (contextual concepts) within the document. Here, the semantic relatedness measure between concepts also adopted link-based concept relatedness method Milne and Witten (2008), Medelyan, Witten, and Milne (2008).

Due to the limited background knowledge and concept mapping technology, extracted concepts might not contain the term information exactly and completely. Many Researchers began to use both term and concept information to represent document, for instance, Term + Concept VSM and Replaced VSM. The Replaced VSM represents document with concepts and terms which do not have any related concept in knowledge base (Wang et al., 2008). Hotho et al. (2003) and Huang et al. (2008) compared three models (concept-based VSM, Term + Concept VSM and Replaced VSM) with term-based VSM. In the experiments, they used the WordNet and Wikipedia as the background knowledge bases respectively. Experimental results showed that Term + Concept VSM usually can improve successfully the performance in text clustering and concept-based VSM did not perform better than term-based VSM in most cases. These observations gave us a hint: concept-based VSM can supply more information for discriminating documents, but only using concepts cannot represent document sufficiently. Concept mapping could result in loss of information or addition of noise. It is necessary to include both term and concept in representation model. In order to make use of term and concept information in text classification and clustering tasks, an alternative method is to liner combining the similarity values which are calculated based on term-based VSM and concept-based VSM respectively (Hu et al., 2008, 2009; Huang et al., 2009; Song, Li, & Park, 2009). However, as shown in the literatures, this method depends on the input parameters.

In this paper, we consider both term-based VSM and concept-based VSM and build a two-level representation model. The term-based VSM is weighted by tf-idf method. In the concept-based VSM, we map term appearing in document to concept via the anchors in Wikipedia. Disambiguation and concept weighting are implemented according to the semantic relatedness between the concept and its context in the document.

As we know, text data is large scale whatever model it is representated in, therefore, it is important to design an effective and efficient learning method to analyze text data. In the literatures, boostrapping and co-training algorithms are suggested to handle large scale text data (Chang, Ratinov, Roth, & Srikuma, 2008). Although boostrapping can improve prediction performance using unlabeled examples, it is still based on a flat feature representation. Co-training algorithm firstly trains two separate classifiers on distinct views and predicts the unlabeled examples respectively. Next, one classifier teaches the other with the most confidential predicted labels of the few unlabeled examples, and these two classifiers are iteratively retrained with the additional training examples separately until all predicated labels do not change (Blum & Mitchell, 1998; Mitchell, 1999). Nigam and Ghani (2000) empirically shows that co-training performs well if the two feature spaces are independent indeed, thus, it can not be directly applied on our proposed 2RM model because term-based feature space and concept-based feature space depends on each other.

In order to efficiently utilize the information recorded in the proposed 2RM model, we design a multi-layer classification framework (MLCLA). MLCLA adopted the idea of co-training, building classifiers on different feature spaces (term-based VSM and concept-based VSM). However, different from co-training method, the proposed MLCLA framework firstly trains two classifiers on different feature spaces in parallel, and then trains the third classifier on the document compressed representation according to the output of the first two classifiers. MLCLA framework does not require that the feature spaces must be sufficient and independent on each other, and such requirement can not be satisfied in real applications. MLCLA is helpful to do mutual complementation between the text information recorded in different feature spaces.

3. Two-level representation model

In this Section, we propose the Two-level Representation Model (2RM) that represents syntactic information and semantic information with two levels. Term-based VSM and tf-idf weighting scheme are used in syntactic level to record the syntactic information. Semantic level consists of Wikipedia concepts related to the terms in the syntactic level. These two levels are connected via the semantic correlation between terms and their relevant concepts. The key technique to build 2RM model is to construct the semantic level. In this paper, a context-based method is proposed to find the most relevant concept for each term based on the document structure information (e.g., document-paragraph) and Wikipedia link structure.

Wikipedia is the largest encyclopedia in the world and useful for natural language processing (Gabrilovich & Markovitch, 2009). The term in Wikipedia article is called as anchor if the term related to a Wikipedia concept, and there will be a hyperlink between the anchor and the concept. The candidate concepts for each term in text corpus can be identified with these hyperlinks (Medelyan et al., 2008). However, there may be twenty concepts related to one term (Hu et al., 2009). Which one or ones are truly semantically related to the term in given document? It is intuitive that term is usually as its several most obvious senses, where the obviousness of sense c for term t is defined as the ratio of the frequency of that t as an anchor linked to c to the frequency of t as an anchor in all the Wikipedia articles (Mihalcea & Csomai, 2007). In this paper, the top η (e.g. $\eta = 3$, 2 or 1) obvious senses are selected as candidate concepts for each term.

The semantic relatedness between term and its candidate concepts in a given document is computed according to the context information as follows.

$$Rel(t, c_i|d_j) = \frac{1}{|T|-1} \sum_{t_l \in T \& t_l \neq t} \frac{1}{|cs_l|} \sum_{c_k \in cs_l} SIM(c_i, c_k) \quad (1)$$

where T is the term set of the jth document d_j, t_l is a term in d_j except for t, cs_l is the candidate concept set related to term t_l. $SIM(c_i, c_k)$ is the semantic relatedness between two concepts, which is calculated with the Wikipedia hyperlinks (Milne & Witten, 2008).

$$SIM(c_i, c_k) = 1 - \frac{log(max(|A|, |B|)) - log(|A \cap B|)}{log(|W|) - log(min(|A|, |B|))} \quad (2)$$

where A and B are the sets of all articles that link to concepts c_i and c_k respectively, and W is the set of all articles in Wikipedia. Eq. (2) is based on term occurrences on Wikipedia-pages. Pages that contain both terms indicate relatedness, while pages with only one of the terms suggest the opposite.

If all terms except for t are taken as t's context in d_j, it will be time-consuming to compute Eq. (1), especially for long document. Therefore, we propose a context identification method based on document structure (e.g., document-paragraph) as shown in Eq. (3). Here, only terms in the predefined window including t are considered as t's contextual information.

$$Rel(t, c_i|d_j, size) = \frac{1}{|T(t, size)|-1} \sum_{t_l \in T(t, size) \& t_l \neq t} \frac{1}{|cs_l|} \times \sum_{c_k \in cs_l} SIM(c_i, c_k) \quad (3)$$

where $T(t, size)$ is the term set appearing in the paragraphs which are the nearest neighbor of the paragraphs term t occurs in

document d_j, and *size* is a the user predefined window size, usually, it is a small integer. For example, when *size* is 1, if t occurs in 3th paragraph, terms in paragraphs 2, 3 and 4 are considered as the context of t. If t occurs in 3th and 7th paragraph, terms in paragraphs 2, 3, 4, 6, 7 and 8 are considered as the context of t. When *size* equals to 0, only the terms in common paragraph are considered as the term's contextual information.

Higher value of $Rel(t, c_i|d_j)$ means that concept c_i is more semantically related to term t, because c_i is much more similar to the relevant concepts of other terms in d_j (such terms are the context of term t). The concepts with highest relatedness will be used to properly build the concept vector in semantic level, i.e., each term will be finally mapped into its most related concept. Based on $Rel(t, c_i|d_j)$ and term's weight $w(t_k, d_j)$, the concept's weight is defined as their weighted sum as follows.

$$w(c_i, d_j) = \sum_{t_k \in T} w(t_k, d_j) \ * \ Rel(t_k, c_i|d_j) \tag{4}$$

Table 1 shows a simple example on how to build 2RM model via the above proposed methods. Given a document, it's important terms are extracted and listed in the first column. Terms' weight values ($tf \cdot idf$) are in Column 3. Column 2 shows the concepts related to the corresponding terms. These concepts have the highest relatedness value $Rel(t, c_i|d_j)$ (in this example, all terms in the document are taken as the contextual information of each term) as shown in Column 4. Column 5 lists the concept weight calculated via Eq. (4) based on Column 3 and Column 4. From Table 1, we can see that different terms may be mapped to a same concept, and some term such as "dealt" has no concept in Wikipedia. Because of these many-to-one mapping, the synonym information can be considered in our proposed 2RM model. In order to deal with the second situation, some terms do not have related concept, a multi-layer classification framework is designed, in the next section, to make use of term and concept information during the classification processing, so that the final classification performance is efficiently proved.

4. Multi-layer classification framework

Multi-layer classification (MLCLA) framework is designed to handle large scale data with complex and high dimensions in a way of layer-by-layer. MLCLA framework consists of two layers with three classifiers for three types of feature spaces. The first two classifiers in lower layer are applied on syntactic level and semantic level independently. Each document will be represented with two compacted vectors according to the similarity between document and all class centers in each classifier. These two compacted vectors are then combined to be the input of the third classifier which will output the final results.

Fig. 1 illustrates the MLCLA framework in detail. In the low layer, the first classifier is trained and tested using the documents which are represented by term-based VSM, i.e., the syntactic information in 2RM model. According to the truth labels of training set and the predicted labels of test set of the first classifier, the center of each class can be determined by averaging the document vectors belonging to this class as showed in Eq. (5).

$$Z_k = \frac{\sum_{d_j \in C_k} d_j}{|C_k|} \tag{5}$$

where $|C_k|$is the number of documents in the kth class C_k. Based on the class centers, each document can be represented with a K-dimension compressed vector $[s_{j1}, \ldots, s_{jK}]$ (K equals to the number of classes) where the value of the kth element is the similarity between document and the kth class center (In our experiments, cosine measure is used here to compute the similarity between document and class center.).

$$s_{jk} = \frac{d_j \bullet Z_k}{\|d_j\| \|Z_k\|} \tag{6}$$

Similarly, the second classifier is applied on the concept-based VSM, i.e., the semantic information in 2RM model, to get the second K-dimension compressed vector $\left[s'_{j1}, \ldots, s'_{jK}\right]$ for each document. Then, two K-dimension compressed vectors are combined as follows.

$$d_j = \left[s_{j1}, \ldots, s_{jK}, s'_{j1}, \ldots, s'_{jK}\right] \tag{7}$$

s_{jk} is the similarity between the jth document represented in syntactic level of the 2RM model and the kth class center obtained by the first classifier. s'_{jk} is the similarity between the jth document represented in semantic level of the 2RM model and the kth class center obtained by the second classifier. This combined document representation will be the input of the third classifier in the high layer of MLCLA, as shown in Fig. 1.

In MLCLA framework, the primary information is effectively kept and the noise is reduced by compressing the original information, so that MLCLA can guarantee the quality of the input of all classifiers. Thus we believe the final classification performance would be improved.

Because MLCLA framework includes two classification procedures in low layer, they can be implemented in series or parallel. When running in series, two data matrices based on different representation levels (syntactic and semantic) can be loaded one by one. Therefore, the required memory space depends on the larger matrix plus compressed representation matrix, rather than the summation of term-based matrix and concept-based matrix. On the other hand, when running in parallel, two classifiers in low layer can be built at the same time, and the classifier in high layer is very fast on the basis of low dimension compression space.

Now we further analyze the time complexity of MLCLA. N represents the number of documents, M denotes the number of terms or concepts in document collection, K is the number of classes, m is the average number of terms or concepts in one document. The low layer of MLCLA includes two classification procedures, based on syntactic level and semantic level respectively. Each classification procedure consists of two parts, document classification and construction of compressed representation. The first one is equal to one basic classifier's time complexity (e.g. O_{SVM}), taking $N \times M$ dimension sparse matrix as input. The second one takes $O(Nm) + O(KNm)$, where computing class centers needs $O(Nm)$ and computing similarities between documents and class centers needs $O(KNm)$. The high layer of MLCLA is equal to one basic classifier's time complexity, taking $N \times 2K$ dimension matrix as input.

Among the existing semantics-based text classification methods, Term + Concept VSM represents each document as a higher dimension vector because it is the mergence of term-based VSM and concept-based VSM. Some kernel-based methods (Jing et al., 2006; Wang et al., 2008) classify documents based on more dense

Table 1
A simple example on two-level representation model for document 20NewsGroup/talk.politics.mideast/76391.

Term (t)	Concept (c)	$w(t,d)$	$R(t,c\|d)$	$w(c,d)$
Sister	Sibling	0.0124	0.0819	1.77e−3
Brother	Sibling	0.0092	0.0819	1.77e−3
Tear	Tears	0.0107	0.0508	9.14e−4
Cry	Tears	0.0073	0.0508	9.14e−4
Fortune	Luck	0.0026	0.0802	4.01e−4
Lucky	Luck	0.0024	0.0802	4.01e−4
Fight	Combat	0.0035	0.0906	3.17e−4
Air	Earth's atmosphere	0.0054	0.0830	4.48e−4
Press	Mass media	0.0016	0.0771	1.23e−4
Party	Political party	0.0018	0.0234	4.21e−5
Dealt	–	0.0031	–	–

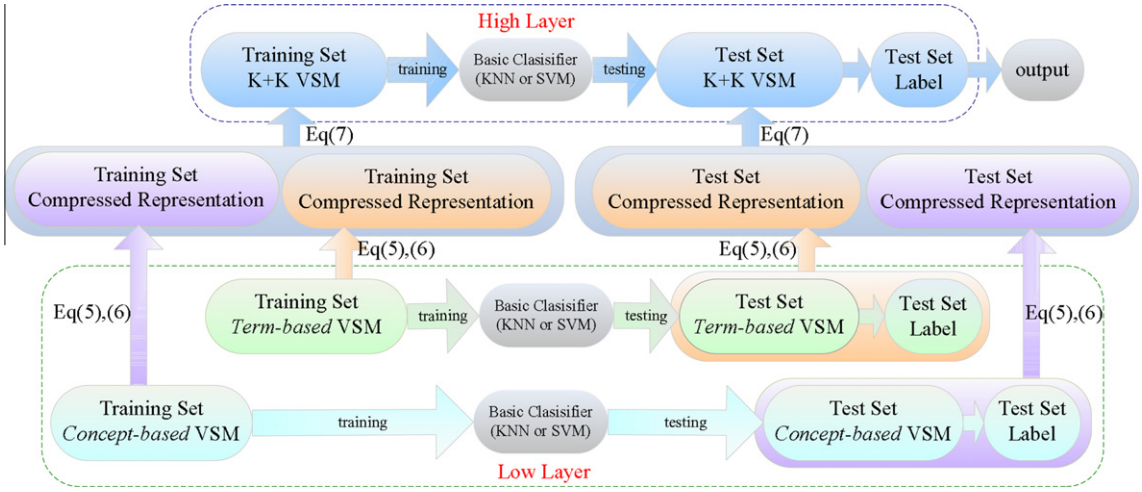

Fig. 1. MLCLA framework.

high dimension matric which is the product of document term (concept) tf-idf matrix and term (concept) correlation matrix. We believe MLCLA would take less space when running in series and less time when running in parallel than these methods. It will be further validated in the experiments in Section 5. Meanwhile, we will also show MLCLA's complexity is not worse than baseline methods even for space in parallel and time in series.

5. Experiments

There are four purposes in the experiments. The main one is to test the performance of proposed 2RM model and MLCLA framework on real datasets by comparing with various flat document representation models plus basic classification algorithm (e.g., SVM or KNN). Besides, in order to tune the unique parameter during building concept vector, how different candidate concept number effects the classification result will be tested. Next, we use a long document dataset to test the time and performance of structure-based relatedness measure method. At last, time and space complexity of MLCLA are analyzed by the experiential data.

5.1. Baseline methods

In order to show the improvement of the proposed approach, we use five flat document representation models including Term-based VSM (T-VSM), Term Semantic Kernel Model (TSKM), Concept-based VSM (C-VSM), Concept Semantic Kernel Model (CSKM) and Term + Concept VSM (TC-VSM) plus one classification algorithm (SVM or KNN) as baseline methods.

- T-VSM means representing document as term vector. Term is weighted by tf-idf method. It has been widely used in information retrieval, text classification and clustering.
- TSKM is designed according to Jing et al. (2006). Semantic information is enriched into term-based VSM by multiplying term correlation matrix. Differently, we construct term correlation matrix on the basis of Wikipedia instead of Wordnet. Formally, TSKM represents document as a term vector. The weight of term is re-calculated by Eq. (8).

$$W_{TSKM}(t_i, d_j) = \sum_{t_l \in T} W_{tfidf}(t_l, d_j) * Rel(t_l, t_i) \quad (8)$$

Where T is the term set in document collection, $W_{tfidf}(t_l, d_j)$ is the tf-idf weight of t_l in d_j. $Rel(t_l, t_i)$ represents the semantic relatedness between two terms t_l and t_i and the relatedness is measured by Eq. (9)

$$Rel(t_l, t_i) = SIM(c_l, c_i), t_l \sim c_l \& t_i \sim c_i \quad (9)$$

Where $t_l \sim c_l$ represents c_l is the most obvious sense of term t_l (the definition of obviousness has been described in Section 3), $SIM(c_l, c_i)$ is introduced in Eq. (2) in Section 3. TSKM implicitly contains the meaning of terms in document, although it still represents document as term vector.

- C-VSM means representing document as Wikipedia concept vector. Concept is extracted and weighted by the method mentioned in Section 3.
- CSKM (Wang et al., 2008) represents document as follows.

$$d_j^{CSKM} = [t_{j1}, \ldots, t_{jN'}, c_{j1}, \ldots, c_{jM}]$$

Where $t_{j1}, \ldots, t_{jN'}$ are the terms which do not have any related concepts in Wikipedia and are weighted by tf-idf. $c_{j1}, \ldots, c_{jM}$ are the concepts related to all the terms in the document set. It is generated by mapping each term to the most obvious sense. It is worth noting that context-based method is not used in CSKM because here concept mapping is run in the whole document collection rather than each document. The weight of concept in document is calculated by Eq. (10)

$$W_{CSKM}(c_i, d_j) = \sum_{c_l \in C} W_{tfidf}(c_l, d_j) * SIM(c_l, c_i) \quad (10)$$

Where C is the concept set in document collection, $W_{tfidf}(c_l, d_j)$ is the tf-idf value of the concept in document. The frequency of the concept c_l in document d_j is equal to the sum of the frequency of terms which are mapped to concept c_l in document d_j. In a word, CSKM first builds document representation vector by complementing concept vector with some terms and then weights concepts by multiplying document concept tf-idf matrix by concept relatedness matrix.

- In TC-VSM, a document is represented by a flat combined vector by appending concepts to term vector. It uses the same weighting method as T-VSM and C-VSM. Classification based on TC-VSM is also equal to combine term and concept information during the learning process by linearly integrating the document similarity based on term vector and similarity based on concept vector. In this case, it can be regarded that the parameter of controling the importance of term or concept is 0.5.

Based on any one of above models, SVM (Scholkopf, Burges, & Smola, 1999) and KNN (Shakhnarovich, Darrell, & Indyk, 2005) were used as basic classifers, where K is set to be 1 in KNN. In this paper, SVM and 1NN algorithms in Weka [4] were used with default

[4] http://www.cs.waikato.ac.nz/ml/weka/.

Table 2
20NewsGroup subsets.

Dataset	Categories
20NG-Binary	talk.politics.mideast, talk.politics.misc
20NG-Multi5	comp.graphics,rec.motorcycles, rec.sport.baseball, sci.space, talk.politics.mideast
20NG-Multi10	alt.atheism, comp.sys.mac.hardware, misc.forsale, rec.autos, rec.sport.hockey, sci.crypt, sci.electronics, sci.med, sci.space, talk.politics.guns
20NG-Diff4	comp.graphics, rec.sport.bassball, sci.space, talk.politics.mideast
20NG-Sim4	comp.graphics, comp.os.ms-windows.misc, rec.autos, sci.electronics
20NG-Long	comp.∗, sci.∗, talk.∗

Table 3
Data set summary.

Dataset	♯Classes	♯Documents	♯Words	♯Concepts	♯Words no Concepts
20NG-Binary	2	500	3376	2987	418
20NG-Multi5	5	500	3310	2735	302
20NG-Multi10	10	500	3344	2772	285
20NG-Diff4	4	4000	5433	4362	503
20NG-Sim4	4	4000	4352	3502	426
20NG-Long	3	210	4244	3738	585
R-Min20Max200	25	1413	2904	2450	176
R-Top 10	10	8023	5146	4109	448
Classic3	3	3891	4745	3737	487

Table 4
F-measure of 4-fold cross validation classification using C-VSM when η is set to 3, 2 and 1.

Dataset	SVM			1NN		
	$\eta = 3$	$\eta = 2$	$\eta = 1$	$\eta = 3$	$\eta = 2$	$\eta = 1$
20NG-Binary	0.8618	0.8617	**0.8858**	0.6359	0.6527	**0.7183**
20NG-Multi5	0.8915	0.9201	**0.9244**	0.5190	0.5795	**0.6054**
20NG-Multi10	0.7848	0.7836	**0.8036**	0.4288	0.3948	**0.4702**
20NG-Diff4	0.9285	0.9300	**0.9398**	0.6510	0.6679	**0.7380**
20NG-Sim4	0.9377	0.9425	**0.9484**	0.5917	0.6079	**0.6540**
R-min20Max200	0.8365	0.8567	**0.8826**	0.4388	0.4572	**0.5212**
R-Top10	0.9104	0.9172	**0.9240**	0.5754	0.5876	**0.6504**
Classic3	0.9894	**0.9918**	0.9908	0.6593	0.7683	**0.8098**

Table 5
F-measure of 4-fold cross validation classification using TC-VSM when η is set to 3, 2 and 1.

Dataset	SVM			1NN		
	$\eta = 3$	$\eta = 2$	$\eta = 1$	$\eta = 3$	$\eta = 2$	$\eta = 1$
20NG-Binary	0.8917	**0.8938**	**0.8938**	0.7307	**0.7408**	0.6960
20NG-Multi5	0.9600	0.9644	**0.9659**	**0.6311**	0.6230	0.6160
20NG-Multi10	0.8826	**0.8847**	0.8798	0.4559	0.4550	**0.4906**
20NG-Diff4	**0.9692**	0.9684	0.9667	0.7579	0.7538	**0.7814**
20NG-Sim4	**0.9875**	0.9869	0.9860	0.6733	0.6812	**0.6991**
R-min20Max200	0.9041	**0.9080**	0.9033	0.5039	0.5200	**0.5385**
R-Top10	**0.9294**	0.9292	0.9272	0.6157	0.6144	**0.6553**
Classic3	**0.9942**	**0.9942**	0.9939	0.8367	0.8351	**0.8468**

parameter value. These baseline models plus basic classifier (SVM or 1NN) will be used to compare with the 2RM plus MLCLA framework. In MLCLA framework, three SVM or three 1NN classifiers were used.

5.2. Datasets

The proposed representation model and multi-layer classification framework were tested on three real data, 20Newsgroups, Reuters-21578 and Classic3. Six subsets were extracted from 20Newsgroups (Jing et al., 2006; Slonim & Tishby, 2000): 20NG-Diff4, 20NG-Sim4, 20NG-Binary, 20NG-Multi5, 20NG-Multi10 and 20NG-Long. Tables 2 and 3 list the categories and the number of documents contained in these subsets.

In this paper, 20NG-Long was created for testing the proposed structure-based concept mapping (Eq. (3)) on fast dealing with long document. It is a collection of long documents containing three categories "comp", "sci" and "talk". In each category, 70 documents with the most large size were extracted from the corresponding topic in 20Newsgroups (documents from topic "rec" were not included because there are few long documents in "rec.∗"). In 20NG-long, the minimal document's size is 10 K, the maximal one is 158 KB and the average size is 29 KB.

Another two data subsets were created from Reuters-21578 following Huang et al. (2009): R-Min20Max200 and R-Top10. R-Min20Max200 consists of 25 categories with at least 20 and at most 200 documents, 1413 documents totally. In R-Top10, 10 largest categories were extracted from the original data set including 8023 doucments. For Classic3, the whole dataset was used in the experiment. In this paper, we only consider the single-label

Table 6
F-measure of 4-fold cross validation classification using 2RM plus MLCLA framework when η is set to 3, 2 and 1.

Dataset	SVM			1NN		
	$\eta = 3$	$\eta = 2$	$\eta = 1$	$\eta = 3$	$\eta = 2$	$\eta = 1$
20NG-Binary	0.9298	0.9258	**0.9278**	0.8577	0.8717	**0.8758**
20NG-Multi5	0.9799	**0.9859**	0.9840	0.9613	**0.9715**	0.9653
20NG-Multi10	0.9130	0.9161	**0.9190**	0.7992	0.8473	**0.8503**
20NG-Diff4	0.9599	**0.9615**	0.9612	0.9447	0.9461	**0.9515**
20NG-Sim4	0.9706	0.9710	**0.9714**	0.9475	0.9506	**0.9554**
R-min20Max200	0.9289	0.9349	**0.9391**	0.8976	0.9239	**0.9450**
R-Top10	**0.9302**	0.9283	0.9253	0.9074	0.9106	**0.9216**
Classic3	0.9936	**0.9943**	0.9938	0.9904	**0.9908**	0.9898

documents. Wikipedia is used as background knowledge which contains 2,388,612 articles (i.e., concepts) and 8,339,823 anchors in English.

From Table 3, we can see the number of words and concepts extracted from each data set. The words were extracted by preprocessing steps, selecting only alphabetical sequences, stemming them, removing stop words and filtering them by the document frequency. Then, we determined the Wikipedia concepts for these words in each document via the method in Section 3 (Note: once a word was stemmed, its original form was used to correctly identify relevant Wikipedia concept). Table 3 shows that the number of distinct concepts appearing in a data set is usually lower than the number of words. Meanwhile, part of words (about 10 percent) do not have relevant concepts.

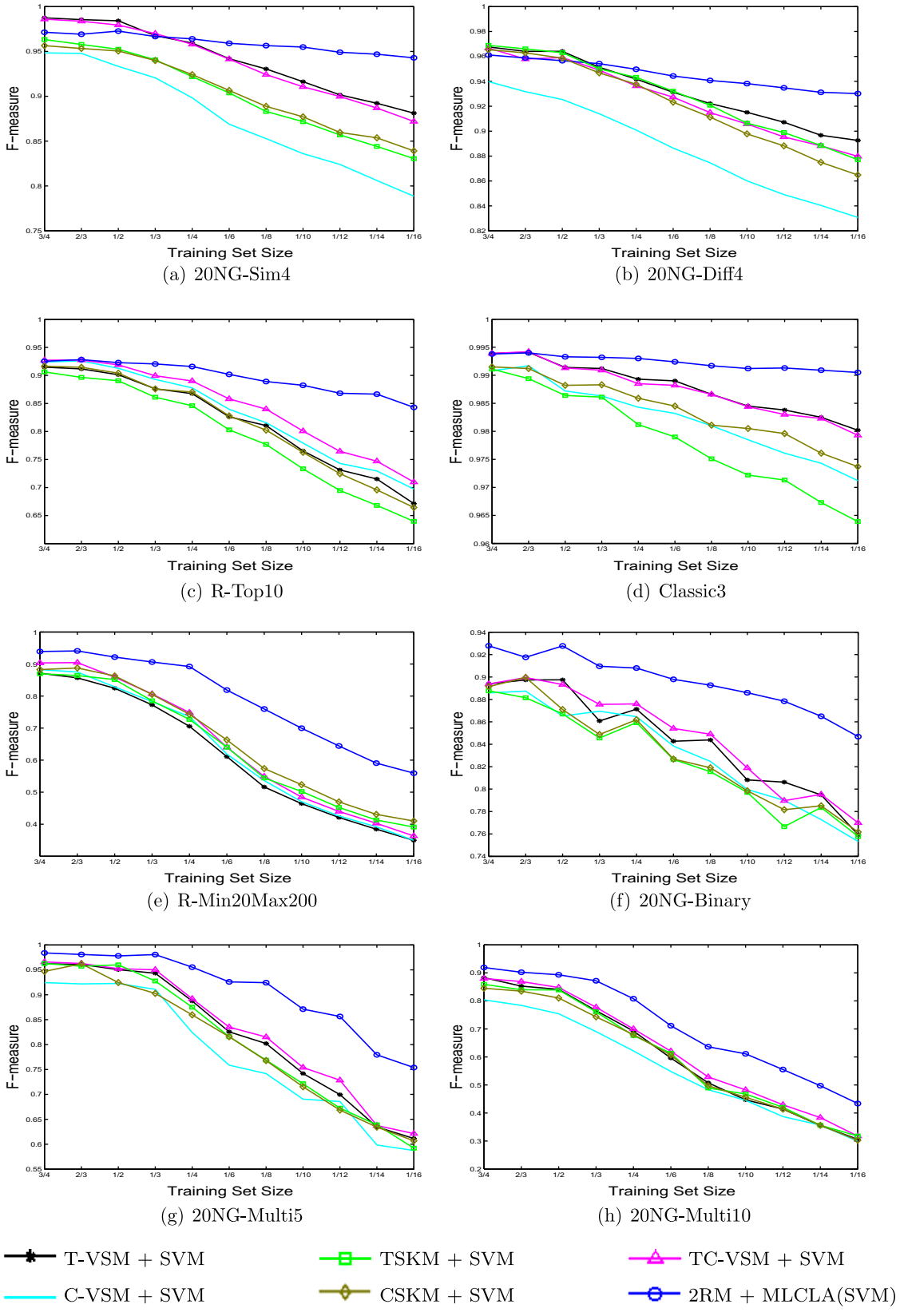

Fig. 2. Compare 2RM plus MLCLA (SVM) with T-VSM, TSKM, C-VSM, CSKM, TC-VSM plus SVM.

5.3. Candidate concept number

In order to find the best concept for each term, we first select top η obvious senses as candidate concepts, and then keep the most related one according to Eq. (1). In order to test how η affects the concept mapping, we compare the classification results when η is set to 3, 2 and 1 respectively. We do three groups of experiments on the basis of two baseline models (C-VSM and TC-VSM) and the

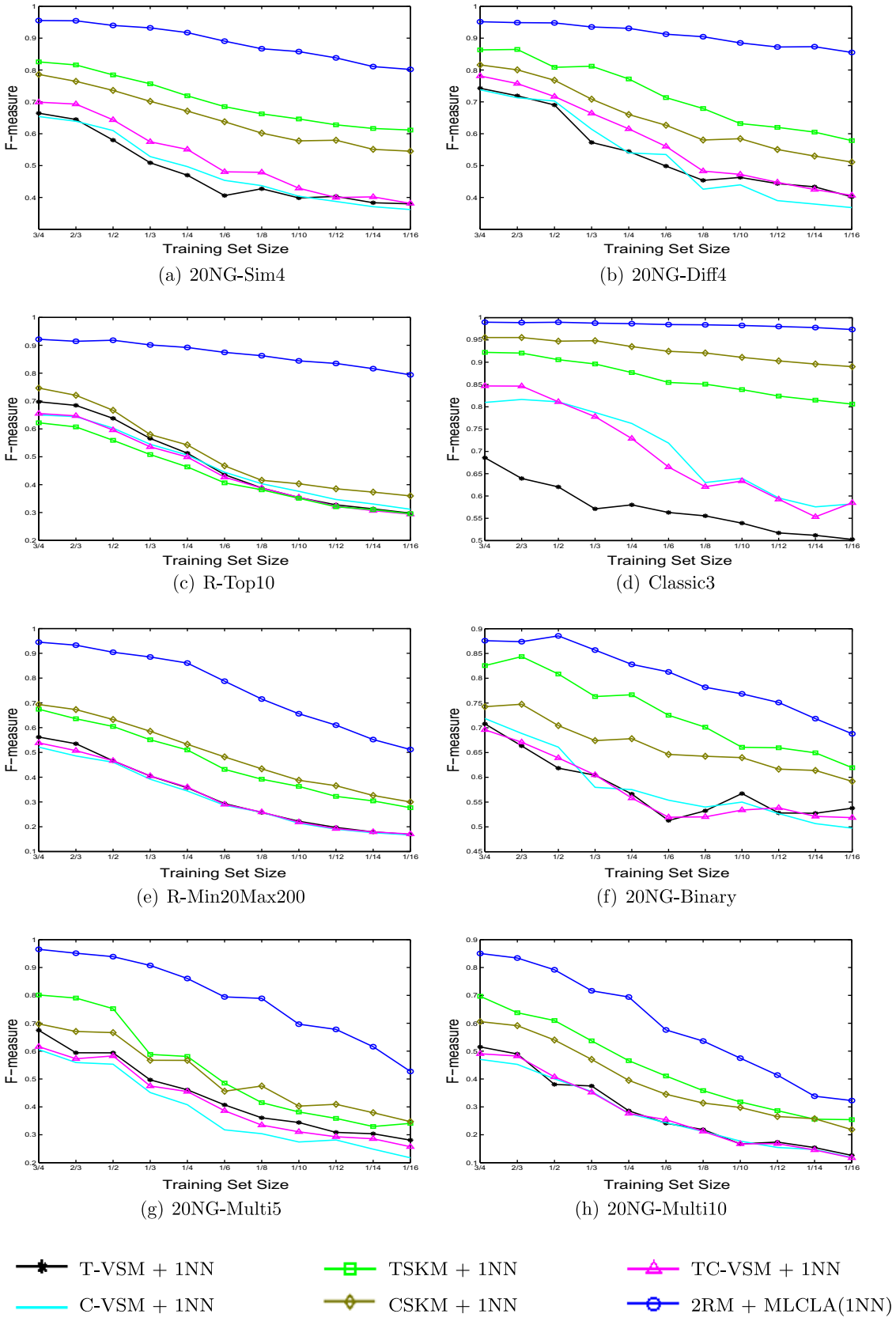

Fig. 3. Compare 2RM plus MLCLA (1NN) with T-VSM, TSKM, C-VSM, CSKM, TC-VSM plus 1NN.

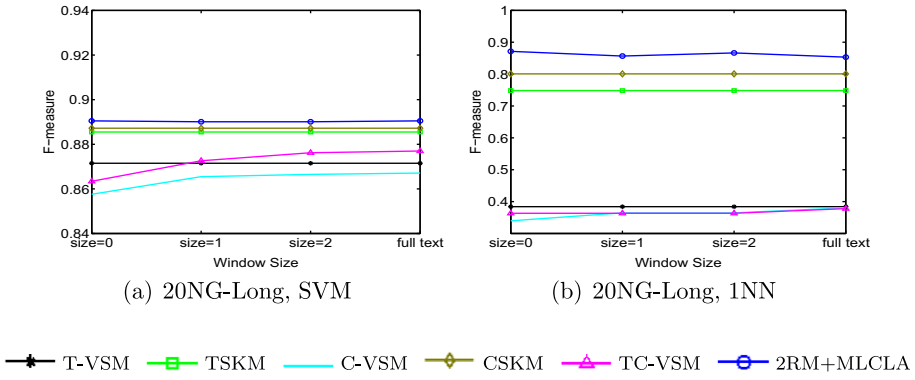

Fig. 4. Compare structure-based relatedness measure with "full text" relatedness measure on 20NG-Long dataset based on (a) SVM and (b) 1NN algorithms respectively.

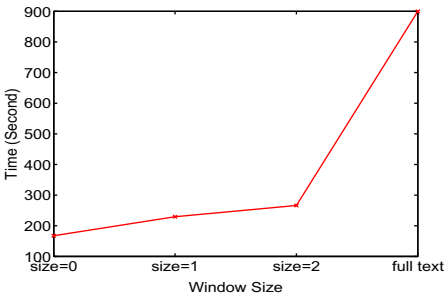

Fig. 5. Time for constructing concept vector by structure-based relatedness measure and "full text" relatedness measure on 20NG-Long dataset.

proposed 2RM, because the parameter η is not mentioned in the other models.

Tables 4–6 list the F-measure of 4-fold cross validation using C-VSM + SVM, C-VSM + 1NN, TC-VSM + SVM, TC-VSM + 1NN, 2RM + MLCLA (SVM) and 2RM + MLCLA (1NN). Bold-face numbers indicate the best evaluation result among different values of η. From Table 4 we can see $\eta = 1$ is the best choose for mapping concept for term in classification when using only concept vector (C-VSM) to represent document. As showed in Tables 5 and 6, TC-VSM and 2RM still get the best results in most dataset for 1NN algorithm when $\eta = 1$, but they achieved very similar performances no matter what η equals for SVM algorithm. This can be explained as TC-VSM or 2RM uses both terms and concepts to represent document, so they are more robust for concept representation. On the other hand, it is evident that smaller η takes less time for constructing the concept vector. Balancing the time against performance, $\eta = 1$ is used in the following experiments.

5.4. MLCLA results

We constructed a series of experiments with different percentages of the training and test data for each dataset (3/4, 2/3, 1/2, 1/3, 1/4, 1/6, 1/8, 1/10, 1/12, 1/14 and 1/16 are the percentages of training data). When the percentage of training data is 3/4, we will use 3/4 of the whole data as training set, while only 1/4 part as training set when the percentage is 1/4, and so on. These experiments would show how the size of training data affects the performance of text classification. The classification performance is measured with F-measure (Feldman & Sanger, 2007) of n-fold cross validation on testing dataset, and higher F-measure value means better classification result. For different sizes of training data, n is set to be different values (4,3,2,3,4,6,8,10,12,14 and 16 respectively). To build the concept vector mentioned in C-VSM, TC-VSM and 2RM, we use "full text" concept mapping (Eq. (1)) rather than structure-based concept mapping (Eq. (3)) in this subsection. The structure-based concept mapping will be tested in long document collection in Section 5.5 in details.

The F-measure curves of SVM and 1NN classifiers with different representation models and different sizes of training data are shown in Fig. 2 (SVM) and Fig. 3 (1NN). From Figs. 2 and 3, we can see that 2RM plus MLCLA framework yields the best results in both SVM and 1NN algorithms, especially it remarkably surpasses the other models when using 1NN as basic classifier. TSKM can get the better results than T-VSM in 1NN algorithm, but fail to improve the performance in SVM algorithm. CSKM achieves the better results than C-SVM. There are two reasons accounting for this improvement. One is different concept weighting strategies, the other is that CSKM uses the Replaced Model which also contains some terms information. TC-VSM is better than TSKM and CSKM in SVM algorithm but worse than these two models in 1NN algorithm.

Meanwhile, the experimental results also demonstrate that only using concepts to represent document (C-VSM) usually gets worse performance than traditional term-based model (T-VSM), because it is difficult to avoid inducing noises and losing information during concept mapping due to the limitation of background knowledge base and word sense disambiguation technique. As popular document representation model with semantic information, TC-VSM's

Table 7
The running time for classification based on different document representation models using SVM classifier (seconds).

Dataset	T-VSM + SVM	C-VSM + SVM	TC-VSM + SVM	TSKM + SVM	CSKM + SVM	2RM + MLCLA (SVM)	
						Series	Parallel
20NG-Binary	0.688	0.515	1.031	2.469	1.89	1.3467	0.8317
20NG-Multi5	0.876	0.734	1.266	1.953	1.25	2.1373	1.4033
20NG-Multi10	2.063	2.094	2.844	4.532	3.985	6.5403	4.4773
20NG-Diff4	17.968	12.327	26.344	31.374	26.407	32.2829	19.9559
20NG-Sim4	17.454	15.454	28.468	76.485	54.564	34.5011	19.0471
R-Min20Max200	12.874	11.359	18.782	28.751	25.718	35.3972	24.0382
R-Top 10	39.031	28.109	61.188	61.469	55.687	75.4846	47.3756
Classic3	8.892	6.016	13.062	23.437	18.47	16.1749	10.1589

Table 8
The running time for classification based on different document representation models using 1NN classifier (seconds).

Dataset	T-VSM + 1NN	C-VSM + 1NN	TC-VSM + 1NN	TSKM + 1NN	CSKM + 1NN	2RM + MLCLA (1NN)	
						Series	Parallel
20NG-Binary	0.687	0.516	1.281	28.5	2.25	1.2966	0.7806
20NG-Multi5	0.609	0.468	1.125	1.578	0.985	1.2751	0.8071
20NG-Multi10	0.577	0.422	1.047	1.719	1.375	1.326	0.904
20NG-Diff4	20.249	16.718	40.938	65.75	52.406	39.3266	22.6086
20NG-Sim4	16.593	14.469	34.281	130.766	109.608	32.9761	18.5071
R-Min20Max200	1.689	1.422	3.469	9.11	7.875	5.2391	3.8171
R-Top 10	32.407	29.282	70.141	77.015	46.656	71.2877	42.0057
Classic3	13.453	12.578	30.735	145.25	83.406	27.5248	14.9468

performances have a little improvement compared with the T-VSM and C-VSM in some cases, but sometimes it also gets much worse results than T-VSM.

In a word, due to the MLCLA framework, 2RM surpasses all the flat vector models and it is robust for the selected basic classification algorithm. In addition, the results shown in Figs. 2 and 3 give us an interesting hint. The 2RM plus MLCLA can get much more improvement when the size of training data is small, especially for large datasets and SVM classifier. This observation is very important because there is always very small number of labeled data (i.e., training data) in real application. Thus, our proposed method seems more useful in practice.

5.5. Structure-based relatedness measure and full text relatedness measure

When building the concept vector space, we use the context-based method to measure the relatedness between term and concept. In this subsection, we compare the fast structure-based relatedness measure (Eq. (3)) with "full text" relatedness measure method (Eq. (1)) by running the experiments in long document collection (20NG-Long dataset). Firstly, we build C-VSM, TC-VSM and 2RM using structure-based relatedness measure method with different paragraph window size (0, 1 and 2) and "full text" relatedness measure method. Next, we do the classification based on various models and compare their F-measures. We also compare the classification results with other three models (T-VSM, TSKM and CSKM), which do not mention the relatedness between term and concept.

Fig. 4(a) and (b) show classification results when SVM and 1NN are used as basic classifier respectively. In *X*-axis, *size* equals to 0, 1 and 2 means the paragraph window size is set to 0, 1 and 2 respectively when using Eq. (3) to compute the relatedness. *full text* represents using all the terms in the whole document as the context (i.e. using Eq. (1) to compute the relatedness). *Y*-axis shows the F-measure of 4-fold cross validation classification. Six representation models are denoted as different curves. From Fig. 4, we can see C-VSM, TC-VSM and 2RM get a little different results when the window size changes. Concretely, the F-measure is higher with the increase of size and the "full text" relatedness measure achieves the best results. However, the change is unremarkable especially for 2RM. In addition, the corresponding classification results of T-VSM, TSKM and CSKM do not vary with the different values in *X*-axis because they do not compute the relatedness between term and concept.

Meanwhile, we also give the time curves for constructing concept vector using different relatedness measure methods in Fig. 5. From Fig. 5 we can see using the proposed structure-based method reduces greatly the time complexity comparing with the "full text" method. This is because the structure-based method decreases greatly the count of concept pairs whose semantic relatedness needs to be calculated when mapping term to concept.

5.6. MLCLA complexity analysis

5.6.1. Time complexity

To build and test the classifier, all the five baseline methods only need to train and test classifier once based on respective representation models. The MLCLA first runs classification task on term-based VSM and concept-based VSM respectively, then computes two compressed representations and trains and tests classifier on the combined compressed representation. Therefore, MLCLA can be executed in series or in parallel. For serial execution, MLCLA runs three classifiers one by one. For parallel execution, the first two classification tasks are run at the same time. Tables 7 and 8 compare the time that these methods take during the classification based on SVM and 1NN respectively. The percentage of training set is 3/4 of the whole data. The time is recorded on the computer with Intel (R) Xeon (R) CPU X3320 2.50 GHz and 4 GB RAM. From Tables 7 and 8, we can see MLCLA in parallel only takes a little more time than the classifiers based on T-VSM and C-VSM, because the first two classifiers in MLCLA are the same as the classifiers based on T-VSM and C-VSM respectively and the third classifier is run on low dimensional feature space which only spends a little time. Meanwhile, MLCLA in parallel takes less time than the other three baseline methods in most cases. On the other hand, MLCLA in series usually takes more time than the classifier based on TC-VSM. But it does not always take more time than the classifier based on TSKM and CSKM. This is due to the sparsity of the document representation matrices in TSKM and CSKM. Especially for 20NG-Binary and 20NG-Sim4 which have similar topics between different categories, many more related terms (concepts) cause more dense correlation matrix. TSKM (CSKM) with more dense document representation matrix need to more time to build and test classifier.

5.6.2. Space complexity

Because all the data are stored in sparse matrix, we compare the space complexity by the number of non zero elements in document representation matrix which are recorded in Table 9. In Table 9, we divide *MLCLA* column into two sub-columns: *series* and *parallel*. MLCLA in series loads term-based data matrix and concept-based matrix one by one, while MLCLA in parallel loads the two data matrices at the same time. Each sub-column contains two parts connected by plus sign. For MLCLA in series, the first part is the number of non zero elements in the larger one of T-VSM matrix and C-VSM matrix. The second part is the number of elements of compression representation matrix which is the output of one of the two classifiers in low layer. For instance, in 20NG-Sim4 there are 16000 elements in the compression representation matrix because the dataset contains 4000 documents and 4 categories (each document would be represented as 4 dimensions compression representation vector). The *series* column shows MLCLA's minimal required space, because MLCLA in series does not need to load term-based data matrix and concept-based data matrix at the same time.

Table 9
The number of non zero elements in different document representation matrices.

Dataset	T-VSM	C-VSM	TC-VSM	TSKM	CSKM	2RM + MLCLA	
						Series	Parallel
20NG-Binary	61447	54799	116246	419781	338460	61447 + 1000	116246 + 2000
20NG-Multi5	43319	39487	82806	169724	89383	43319 + 2500	82806 + 5000
20NG-Multi10	41263	37925	79188	181503	140429	41263 + 5000	79188 + 10000
20NG-Diff4	325209	300523	625732	1130362	847714	325209 + 16000	625732 + 32000
20NG-Sim4	276860	255126	531986	3193044	2371254	276860 + 16000	531986 + 32000
R-Min20Max200	61786	59370	121156	502546	426695	61786 + 35325	121156 + 70650
R-Top 10	263033	253228	516261	661489	411310	263033 + 80230	516261 + 160460
Classic3	185960	178405	364365	3269951	1781318	185960 + 11673	364365 + 23346

In another words, although MLCLA takes more memory space in total, we can train and test the two classifiers in low layer one by one if the memory space is very limited. Table 9 shows MLCLA in series takes less memory space than TC-VSM, TSKM and CSKM. We can also see MLCLA in parallel takes less memory space than TSKM and CSKM. TSKM and CSKM models occupy huge memory space is because they enrich document representation by the relatedness between terms (concepts). The high memory usage of TSKM and CSKM models is more serious for some datasets which contain similar topics, such as 20NG-Binary and 20NG-Sim4, because they cover much more related terms (concepts).

6. Conclusion and future work

In this paper, we represent document as a two-level model with the aid of Wikipedia. In the two-level representation model, one for term information, the other for concept information and these levels are connected by the semantic relatedness between terms and concepts. A context-based method is adopted to identify the relatedness between terms and concepts by utilizing the link structure among Wikipedia articles, which is also used to select the most appropriate concept for a term in a given document. Furthermore, we also propose a fast structure-based relatedness measure to reduce the time of concept mapping. Based on the two-level representation model, we propose a multi-layer classification (MLCLA) framework to analyze text data. Experimental results on real data sets (20Newsgroups, Reuters-21578 and Classic3) have shown that the proposed model and classification framework significantly improved the classification performance by comparing with the existing flat vector models plus the traditional classification algorithms. Besides, MLCLA can be implemented flexibly to run in series or in parallel. We experimentally show MLCLA takes less time when running in parallel and less space when running in series.

In the future, we will focus on the concept mapping and weighting technology to find the better concept vector space for documents, because the better concept-based representation can help to further improve the performance of multi-layer classification framework. Moreover, we will also exploit a new semantic-based vector space model utilizing the category information in Wikipedia. Afterward, two-level representation model will be extended to three-level model containing term, concept and category information respectively. At last, MLCLA will also be improved to fit the three-level model. With the aid of the rich background knowledge, we believe new MLCLA will achieve more predominant classification performance.

Acknowledgments.

We thank David Milne for providing us with the Wikipedia Miner tool-kit. This work was supported in part by the National Natural Science Foundation of China (60905028, 90820013, 60875031), and the National Grand Fundamental Research 973 Program of China under Grant (Nos. 2007CB311002, 2007CB307100, 2007CB307106)

References

Banerjee, S., Ramanathan, K., & Gupta, A. (2007). Clustering short texts using wikipedia. In *Proceedings of the 30th ACM SIGIR* (pp. 787–788).
Blei, D., Ng, A., & Jordan, M. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research, 3*, 993–1022.
Blum, A., & Mitchell, T. (1998). Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th annual conference on computational learning theory* (pp. 92–100).
Chang, M., Ratinov, L., Roth, D., & Srikuma, V. (2008). Importance of semantic representation: Dataless classification. In *Proceedings of the 23rd AAAI conference on artificial intelligence* (pp. 830–835).
Chow, T., & Rahman, M. (2009). Multilayer some with tree-structured data for efficient document retrieval and plagiarism detection. *IEEE Transactions on Neural Networks, 20*, 1385–1402.
Deerwester, S., Dumais, S., Landauer, T., Furnas, G., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the Society for Information Science, 41*(6), 391–407.
Feldman, R., & Sanger, J. (2007). *The text mining handbook: Advanced approaches in analyzing unstructured data*. Cambridge University Press.
Gabrilovich, E., & Markovitch, S. (2005). Feature generation for text categorization using word knowledge. In *Proceedings of the 19th international joint conference on artificial intelligence, Edinburgh* (pp. 1048–1053).
Gabrilovich, E., & Markovitch, S. (2006). Overcoming the brittleness bottleneck using wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the 21st AAAI. Boston, MA, USA* (pp. 1606–1611).
Gabrilovich, E., & Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th IJCAI* (pp. 1606–1611).
Gabrilovich, E., & Markovitch, S. (2009). Wikipedia-based semantic interpretation for natural language processing. *Journal of Artificial Intelligence Research, 34*, 443–498.
Hotho, A., Staab, S., & Stumme, G. (2003). Wordnet improves text document clustering. In *Proceedings of the semantic web workshop at the 26th ACM SIGIR* (pp. 541–544).
Hu, J., Fang, L., Cao, Y., Zeng, H., Li, H., Yang, Q., & Chen, Z. (2008). Enhancing text clustering by leveraging wikipedia semantics. In *Proceedings of the 31st ACM SIGIR* (pp. 179–186).
Hu, X., Zhang, X., Lu, C., Park, E., & Zhou, X. (2009). Exploiting wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD* (pp. 389–396).
Huang, A., Milne, D., Frank, E., & Witten, I. (2008). Clustering documents with active learning using wikipedia. In *Proceedings of international conference on data mining series* (pp. 839–844).
Huang, A., Milne, D., Frank, E., & Witten, I. (2009). Clustering documents using a wikipedia-based concept representation. In *Proceedings of the 13rd PAKDD* (pp. 628–636).
Jing, L., Zhou, L., Ng, M., & Huang, J. (2006). Ontology-based distance measure for text clustering. In *Proceedings of the 4th workshop on text mining, the 6th SIAM international conference on data mining*.
Medelyan, O., Witten, I., & Milne, D. (2008). Topic indexing with wikipedia. In *Proceedings of the AAAI wikipedia and AI workshop*.
Mihalcea, R., & Csomai, A. (2007). Wikify! linking documents to encyclopedic knowledge. In *Proceedings of the 16th CIKM* (pp. 233–242).
Milne, D., & Witten, I. (2008). An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of the workshop on Wikipedia and artificial intelligence at AAAI* (pp. 25–30).
Mitchell, T. (1999). The role of unlabeled data in supervised learning. In *Proceedings of the 6th international colloquium on cognitive science*.
Nigam, K., & Ghani, R. (2000). Analyzing the effectiveness and applicability of co-training. In *Proceedings of the 9th international conference on information and knowledge management* (pp. 86–93).

Nouali, O., & Blache, P. (2004). A semantic vector space and features-based approach for automatic information filtering. *Expert Systems with Applications, 26*(2), 171–179.

Scholkopf, B., Burges, C., & Smola, A. (1999). *Advances in kernel methods: Support vector learning*. Cambridge, MA: MIT Press.

Shakhnarovich, G., Darrell, T., & Indyk, P. (2005). *Nearest-neighbor methods in learning and vision*. The MIT Press.

Slonim, N., & Tishby, N. (2000). Document clustering using word clusters via the information bottleneck method. In *Proceedings of 23th ACM SIGIR* (pp. 208–215).

Song, W., Li, C., & Park, S. (2009). Genetic algorithm for text clustering using ontology and evaluating the validity of various semantic similarity measures. *Expert Systems with Applications, 36*(5), 9095–9104.

Syed, Z., Finin, T., & Joshi, A. (2008). Wikipedia as an ontology for describing documents. In *Proceedings of the 2nd international conference on weblogs and social media, Washington* (pp. 136–144).

Wang, P., & Domeniconi, C. (2008). Building semantic kernels for text classification using wikipedia. In *Proceedings of the 14th ACM SIGKDD. New York, NY, USA* (pp. 713–721).

Wang, P., Hu, J., Zeng, J., Chen, L., & Chen, Z. (2007). Improving text classification by using encyclopedia knowledge. In *Proceedings of the 7th ICDM. Omaha, NE, USA* (pp. 332–341).

Yates, R., & Neto, B. (1999). *Modern information retrieval*. Addison-Wesley.